

PROGRAM-CONTROLLED UNIT

5 Cross-Reference to Related Application:

This application is a continuation of copending International Application No. PCT/DE99/01988, filed July 1, 1999, which designated the United States.

10 Background of the Invention:

Field of the Invention:

15 The invention lies in the computer technology field and pertains, more specifically, to a program-controlled unit with one or more application-specifically configurable, intelligent interfaces.

20 Program-controlled units are devices controlled by software programs, that is to say microprocessors, microcontrollers or the like. They exist in a multiplicity of diverse embodiments and need not be described in specific detail.

25 Even though microprocessors, microcontrollers and the like can be obtained in a wide variety of embodiments (or perhaps precisely because of this), it is difficult or sometimes even impossible to find one type which can satisfy the requirements

imposed in the specific individual case (and as far as possible only these requirements).

This is particularly applicable to microcontrollers because after all, if possible, all the peripherals (analog/digital converter, digital/analog converter, timer, interrupt controller, etc.) required for the relevant application are intended to be integrated in the microcontrollers.

If the microcontroller selected for the relevant application is one which - for whatever reason - cannot satisfy all of the requirements imposed, then this generally means that the hardware within which the microcontroller is intended to be used becomes more complex. This turns out to be particularly difficult and complex in the case of microcontrollers because the latter are not designed to cooperate with coprocessors and the like.

On the other hand, if the microcontroller selected for the relevant application is one which can do more than is required, then the price of the product containing the microcontroller generally rises as a result. Added to this is the fact that often the operation of the microcontroller also becomes more complex and more complicated. This is because the unrequired peripheral units of the microcontroller also have to be taken into account during the system development and be

controlled as intended by the intelligent core of the microcontroller (the so-called microprocessor core or μ P core) which processes the instructions to be executed.

5 Summary of the Invention:

It is accordingly an object of the invention to provide a program-controlled unit, which overcomes the above-mentioned disadvantages of the heretofore-known devices and methods of this general type and which provides for a program-controlled unit that can be used optimally in each case for a large number of highly varied applications.

With the foregoing and other objects in view there is provided, in accordance with the invention, a program-controlled unit with one or more application-specifically configurable intelligent interfaces, comprising:

an intelligent core configured to process instructions to be executed;

a plurality of units selected from the group consisting of internal peripheral units disposed inside the program-controlled unit, external peripheral units exterior to the program-controlled unit, and one or more memory devices;

a structurable hardware unit selectively forming an application-specifically configurable intelligent interface

for respectively connecting the intelligent core and the units, including an interface connection between the intelligent core and the internal peripheral units, between the intelligent core and the external peripheral units, between the intelligent core and the memory devices, and between the plurality of units; and

wherein the structurable hardware unit is configured to evaluate and process data and/or signals received thereby.

Alternatively, or in addition, the structurable hardware unit is configured to inject instructions into an instruction pipeline of the intelligent core.

In another alternative, or in addition, the structurable hardware unit is configured to generate and to output interrupt requests and/or event-signaling messages.

Finally, in a further alternative, the structurable hardware unit is configured to selectively react to interrupt requests or other event-signaling messages from devices connected thereto and prevent the interrupt requests or the event-signaling messages from being forwarded.

In sum, the program-controlled unit comprises one or more application-specifically configurable intelligent interfaces.

Given appropriate configuration, such interfaces can to the greatest possible extent independently ensure that the intelligent core and one or more peripheral units and/or the intelligent core and one or more memory devices and/or two or more peripheral units themselves and/or one or more peripheral units and one or more memory devices cooperate as desired with minimal loading of the intelligent core.

10 In accordance with an added feature of the invention, the structurable hardware unit is disposed in circuit terms between the intelligent core and the plurality of units.

15 In accordance with an additional feature of the invention, the structurable hardware unit is connected to a multiplicity of potential data and signal sources and data and signal destinations, and a plurality of multiplexers are connected to the structurable hardware unit for selecting current data and signal sources and current data and signal destinations.

20 In accordance with another feature of the invention, the data and signal sources and the data and signal destinations comprise the intelligent core, the peripheral units, the memory devices, and/or portions of the structurable hardware unit itself.

25

In accordance with a further feature of the invention, the structuring of the structurable hardware unit selectively results in an alteration of given data paths and/or in a configuration of logic elements.

5

In accordance with again an added feature of the invention, the structurable hardware unit comprises a clock generation unit generating a clock signal and a logic block unit connected to receive the clock signal, the logic block unit enabling devices to be connected via the structurable hardware unit to cooperate as desired.

In accordance with again an additional feature of the invention, the clock generation unit and the logic block unit each contain configurable elements.

In accordance with again another feature of the invention, the clock generation unit is formed at least in part by a DNF logic configuration, a NAND array, a multiplexer-based logic variant, and/or a structurable logic configuration.

In accordance with again a further feature of the invention, the logic block unit is formed at least in part by a DNF logic configuration, a NAND array, a multiplexer-based logic variant, and/or a structurable logic configuration.

In accordance with yet an added feature of the invention, the logic block unit comprises at least one logic block subdivided at least partly into individually configurable sub-blocks with predetermined tasks. Preferably, one of the sub-blocks is

5 configured as a processing device enabled for arithmetic and/or logical processing of data input to the sub-block. It is also possible for one of the sub-blocks to be configured as a state machine for central sequence control. Similarly, one of the sub-blocks may be configured as an address calculation

10 device for calculating source and destination addresses. Also, it is possible for one of the sub-blocks to be configured as an instruction injection device for injecting instructions into the instruction pipeline of the intelligent core.

15 In accordance with yet an additional feature of the invention, the structurable hardware unit is configurable with fuses and/or anti-fuses.

In accordance with yet another feature of the invention, the

20 structurable hardware unit is reversibly configurable. In a preferred embodiment, the structurable hardware unit is configurable based on data representing a desired configuration, and the data are stored in memory devices insertible into a memory or I/O area which is addressible by

25 the intelligent core.

In accordance with yet a further feature of the invention, the structurable hardware unit is enabled for reconfiguration only at predetermined times.

- 5 In accordance with a concomitant feature of the invention, the structurable hardware unit is enabled for reconfiguration at any time.

10 If the interfaces are made in such a way that they are reconfigurable during entirely normal operation of the program-controlled unit, then the program-controlled unit can even be reconfigured dynamically.

15 Consequently, a program-controlled unit has been created which can be used optimally in each case for a large number of highly varied applications.

Other features which are considered as characteristic for the invention are set forth in the appended claims.

20

Although the invention is illustrated and described herein as embodied in a program-controlled unit, it is nevertheless not intended to be limited to the details shown, since various modifications and structural changes may be made therein

25 without departing from the spirit of the invention and within the scope and range of equivalents of the claims.

The construction and method of operation of the invention, however, together with additional objects and advantages thereof will be best understood from the following description
5 of specific embodiments when read in connection with the accompanying drawings.

Brief Description of the Drawings:

Fig. 1 is a schematic illustration of the construction of the
10 program-controlled unit according to the invention;

Fig. 2 is a schematic illustration of the construction of an
SLE layer 12 of the program-controlled unit in accordance with
Fig. 1;

Fig. 3 is a schematic diagram of the construction of a clock
generation unit 121 of the SLE layer 12 in accordance with
Fig. 2;

Fig. 4 is a schematic diagram of the construction of a logic
block unit 122 of the SLE layer 12 in accordance with Fig. 2;
and

Fig. 5 is a schematic diagram illustrating the configuration
25 of the logic block unit 122 in accordance with Fig. 4 for a practical example.

Description of the Preferred Embodiments:

The exemplary program-controlled unit described in more detail below is a microcontroller. It will be understood, however, that the program-controlled unit can, in principle, also be any other program-controlled unit, such as a microprocessor for example.

The microcontroller considered herein is designated by the reference symbol 1 in the figures. It is distinguished by the fact that it comprises one or more application-specifically configurable intelligent interfaces.

The novel microcontroller construction is referred to below as application-specifically structurable controller architecture or ASSC architecture.

The general structure of a microcontroller having the ASSC architecture is schematically illustrated in Fig. 1.

Referring now to the figures of the drawing in detail and first, particularly, to Fig. 1 thereof, there is shown an intelligent core 11, which is usually referred to as microprocessor core or μ P core and processes the instructions to be executed, a so-called SLE layer 12 and also peripheral units 13 to 19, the peripheral units, for their part,

comprising a serial interface unit 13, a parallel interface unit 14, an analog/digital converter 15, a digital/analog converter 16, a timer 17, an interrupt controller 18 and, if appropriate, further peripheral units 19. Furthermore, an
5 external random access memory (RAM) 2 and an external read-only memory (ROM) 3 are provided; the RAM 2 and the ROM 3 can also be realized as internal memories without any difficulty.

The SLE layer 12 is arranged in circuit terms between the μ P core 11, peripheral units (for example the peripheral units 13 to 19) provided inside and/or outside the program-controlled unit, and/or memory devices (for example the RAM 2 and/or the ROM 3). It contains structurable data paths and/or logic elements which can be structured or configured in such a way
10 that the SLE layer 12 can be used as the at least one application-specifically configurable interface, more precisely as a configurable intelligent interface between the μ P core and one or more peripheral units and/or between the μ P core and one or more memory devices and/or between two or more
15 peripheral units themselves and/or between one or more peripheral units and one or more memory devices.
20

Within the SLE layer 12 there are preferably both direct connections and configurable data paths and data path linkages
25 between the devices which are connected or can be connected

via the SLE layer; the effect achieved by the direct connections is that the novel microcontroller can also be utilized like a conventional microcontroller (not having an SLE layer 12).

5

If the SLE layer 12 has access to memory devices (for example to the RAM 2 and/or to the ROM 3) it can transfer data from and to the memory devices independently, i.e. without the participation of the μ P core, for itself and/or the μ P core and/or the peripheral units. As a result, data transfers can be carried out more rapidly and without burdening the μ P core 11. In this case, the SLE layer 12 is designed in such a way that it can identify and handle contending accesses to memory devices.

15

The SLE layer 12 can also be constructed without an interface to the memory devices present. In that case, the SLE layer and the μ P core are preferably designed in such a way that the SLE layer can inject into the μ P core (more precisely into the instruction pipeline thereof) instructions which can cause the μ P core to carry out data transfers required by the SLE layer and/or the peripheral units.

20

The basic construction of the SLE layer 12 is illustrated in Fig. 2. The SLE layer 12 comprises a clock generation unit 121 and a logic block unit 122.

5 The clock generation unit 121 serves for supplying the logic block unit 122 with one or more different clock signals OUTCLK generated on the basis of one or more master clock signals INCLK. By way of example, clock signals having an altered frequency, altered duty ratio and/or altered phase angle relative to the master clock signal or signals INCLK are generated in the clock generation unit 121.

10 In the example considered, the clock generation unit 121 comprises a fast configurable logic arrangement, for example a so-called DNF logic arrangement (NOT-AND-OR logic), suitable for the practical realization of a disjunctive normal form, with downstream configurable storage, inversion/buffering and feedback. A two-stage DNF architecture (having two DNF blocks) is used in the example considered.

20

The fundamental construction of a clock generation unit 121 which generates an output clock signal OUTCLK on the basis of a master clock signal INCLK (for example the CPU clock signal) is illustrated in Fig. 3. The customary form of representation for PALs (programmable array logic) and GALs (Generic array logic) has been chosen; transparent nodes represent

configurable connections, and filled-in (black) nodes represent permanent connections.

In the example considered, the clock generation unit 121 comprises two identically constructed DNF blocks DNF1 and DNF2, the first DNF block DNF1 representing a first DNF logic stage and comprising AND elements A1, an OR element O1, a flip-flop FF1 (clocked by the master clock signal), and a multiplexer MUX1, and the second DNF block DNF2 representing a second DNF logic stage and comprising AND elements A2, an OR element O2, a flip-flop FF2 (clocked by the master clock signal), and a multiplexer MUX2. The two DNF blocks DNF1 and DNF2 are supplied with the same input signals. In the example considered, the input signals comprise the master clock signal INCLK, the inverted master clock signal /INCLK, an output signal OUT1 of the first DNF block DNF1, the inverse output signal /OUT1, an output signal OUT2 of the second DNF block DNF2, and the inverse output signal /OUT2. The output signal OUT2 of the second DNF block DNF2 and the inverse output signal /OUT2 are fed to a multiplexer MUX3, and the output signal thereof is output via a driver T2 as the output clock signal OUTCLK to be generated.

The input signals, more precisely signal combinations which can be selected via configurable connections, applied to the respective DNF blocks DNF1 and DNF2 are applied to the AND

elements A1 and A2 of the respective DNF blocks. The output signals of the AND elements A1 and A2 are fed to the OR elements O1 and O2, respectively, or used as a reset signal ASR for asynchronously resetting the flip-flops FF1 and FF2, respectively. The output signals of the OR elements O1 and O2 are used as input signals of the flip-flops FF1 and FF2, respectively; they (the output signals of the OR elements O1 and O2) and the inverse output signals are additionally applied to the multiplexers MUX1 and MUX2, respectively. The output signals and the inverse output signals of the flip-flops FF1 and FF2 are likewise applied as input signals to the multiplexers MUX1 and MUX2, respectively. The output signal and the inverse output signal of the multiplexers MUX1 and MUX2 are the output signals OUT1, /OUT1, OUT2 and /OUT2, already mentioned above, of the two DNF blocks of the clock generation unit 121.

Such a construction, or a similar construction, of the clock generation unit 121 allows the generation of a multiplicity of clock variations; the feedback of (internal) signals (the signals OUT1, /OUT1, OUT2, /OUT2 in the example considered) generated within the clock generation unit 121 also makes it possible to realize non-binary duty and division ratios.

The arrangement shown in Fig. 3 is only a simple example for explaining the fundamental construction of the clock

generation unit 121. In practice, the number of clock signals generated by the clock generation unit 121 should not be less than four. The lower limit for the possible number of terms per DNF should likewise be four. For each generated clock
 5 signal, at least one internal signal should be generated and be provided like the internal signals OUT1, /OUT1, OUT2 and /OUT2, for example, as a possible basis for the clock signal generation.

10 The clock generation unit 121 need not necessarily be constructed using a two-stage DNF logic arrangement. The DNF logic arrangement used may also have fewer stages (that is to say just one stage) or arbitrarily many more stages.

15 Instead of the DNF logic arrangement, it is also possible to use another structurable logic arrangement, for example NAND arrays or multiplexer-based variants.

The logic block unit 122 already mentioned above is the actual
 20 core of the SLE layer 12. Like the clock generation unit 121, it may comprise a structurable logic arrangement suitable for the practical realization of the disjunctive normal form, or one of the alternatives mentioned. Virtually any desired linkages, processing and evaluations of the input signals can
 25 be carried out by means of such universally usable

structurable logic arrangements. A logic block unit 122 of this type is extremely flexible.

In the example considered in the present case, the logic block unit 122 has been realized differently: it contains one or more logic blocks which are subdivided at least partly into sub-blocks with predetermined tasks. This simplifies the complexity of the gates used but hardly reduces the flexibility of the SLE layer 12 - at any rate given a suitable definition of the sub-blocks and corresponding cross-connections between the individual sub-blocks.

The fundamental construction of a logic block subdivided into various sub-blocks is illustrated in Fig. 4. The logic block shown therein comprises four sub-blocks, namely

- a first sub-block 5 for the inputting and/or outputting of data and/or signals and the processing thereof,
- a second sub-block 6 for central sequence control for the operations proceeding within the relevant logic block,
- a third sub-block 7 for address calculations, and
- a fourth sub-block 8 for instruction injections into the μP .

In the example considered, the sub-blocks comprise configurable multiplexers (for switching data paths as desired), registers (for buffer-storing data and/or coded states) and structurable logic (for linking data and/or signals with one another and with constants and for coding and decoding states).

In the example considered, the first sub-block 5 serves for linking input data and/or signals with one another or with constants; in the example considered, it comprises a first multiplexer 51, a second multiplexer 52, a constant register 53, a first structurable logic arrangement 54, a second structurable logic arrangement 55, and a register 56.

Data and/or signals switched through by the first multiplexer 51 are linked with one another or with constants stored in the constant register 53 by the structurable logic arrangements 54 and 55 and output via the second multiplexer 52. The register 56 provided between the structurable logic arrangements 54 and 55 serves as a buffer store which can be used, for example, for buffer-storing data that are to be linked until the other data with which they are to be linked are valid; the data and/or signals output by the structurable logic arrangement 54 can alternatively be transmitted via a bypass directly

(without a detour via the register 56) to the structurable logic arrangement 55.

The multiplexers 51 (source selection) and 52 (destination selection) connect the first sub-block 5 to different data and/or signal sources and data and/or signal destinations. Data and/or signal sources and data and/or signal destinations are, in particular, the other sub-blocks, the μ P core 11, peripheral units (e.g. the peripheral units 13 to 19) provided inside and/or outside the microcontroller, and/or memories (e.g. the RAM 2 and/or the ROM 3). The connection to the aforementioned or other data and/or signal sources and data and/or signal destinations can be effected via databuses (but does not have to be).

The first sub-block 5 is operatively comparable with the arithmetic logic unit (ALU) of a program-controlled unit working according to the Von Neumann Principle.

In the example considered, the second sub-block is a state machine for central sequence control in the relevant logic block; in the example considered, it comprises a first structurable logic arrangement 61, a second structurable logic arrangement 62, a register 63, and a multiplexer 64.

It is incumbent upon the structurable logic arrangement 61 to determine the respective state of the relevant logic block, in particular in a manner dependent on the states, signals and signal profiles in or from the clock generation unit 121, the other sub-blocks, the μ P core 11, the peripheral units (e.g. the peripheral units 13 to 19) and/or the memories (e.g. the RAM 2 and/or the ROM 3); the operation which should (must) currently proceed in the logic block depends on the respective state of said logic block. The aforesaid state, more precisely a data value representing it, may be stored in the register 63 and be decoded as required by the structurable logic arrangement 62. The structurable logic arrangement 62 outputs one or more output signals. These output signals are control signals based on the respective state of the relevant logic block and are output via the multiplexer 64 in particular to the other sub-blocks, but also, if appropriate, to the μ P core, the peripheral units and/or the memory devices.

In the example considered, the second sub-block 6 can be used to realize different simple automata types (up to Mealy), to be precise also with different clocks.

The task incumbent upon the second sub-block 6 is comparable with the control unit of a Von Neumann CPU, the processing of

instructions naturally being omitted (the states are traversed in a manner triggered by external signals).

In the example considered, the third sub-block 7 serves in particular (but not exclusively) for address calculation for block-by-block data transfers; in the example considered, it comprises an address register 71, a constant register 72, an incrementing/decrementing unit 73, a first structurable logic arrangement 74 and a second structurable logic arrangement 75.

The third sub-block 7 can calculate source and destination addresses and output them to the other sub-blocks, the μ P core, the peripheral units and/or the memory.

The present address in each case is stored in the address register 71 and can be incremented or decremented by the incrementing/decrementing unit 73. The constant register 72 is designed to store an end address, and an address comparison can be carried out by the structurable logic arrangements (in order, for example, to be able to check whether the address stored in the address register 71 has reached the end address stored in the constant register 72).

The provision of this third sub-block 7 is optional and is very useful in particular (but not exclusively) when the SLE layer 12 has direct access to the memory.

In the example considered, the fourth sub-block 8 serves for injecting instructions into the pipeline of the μ P core 11; in the example considered, it comprises a constant register 81 and a structurable logic arrangement 82.

Instructions are injected under the control of the second sub-block 6, more precisely the control signals output by the latter. The instruction code of the instruction to be injected is stored in the constant register 81 and is communicated as required (if appropriate together with address and data information) via the structurable logic arrangement 82 to the μ P core 11; for this purpose, the second sub-block 6 puts the structurable logic arrangement 82, as required, into a state in which the instruction code stored in the constant register 81 (if appropriate together with the data output from the first sub-block 5 and/or the address output from the third sub-block 7) is injected into the pipeline of the μ P core.

As has already been mentioned or indicated at least in part above, there are diverse cross-connections between the individual sub-blocks 5 to 8. The cross-connections depicted in Fig. 4 are to be regarded only as examples. The number of cross-connections and the beginning and end points thereof depend in particular on the concrete application of the

relevant logic block and the number and function of the sub-blocks thereof and may be designed, as required, as configurable connections.

- 5 The at least partial division of the logic blocks of the logic block unit 122 of the SLE layer 12 into sub-blocks enables the practical realization of the logic blocks with minimal complexity. In particular, considerably fewer linkage possibilities have to be provided compared with the realization of the logic block unit using one or more structurable logic arrangements which can be used universally; the reason for this is that because each sub-block only has to perform one precisely defined task, the sub-blocks in each case have to be able to link only a few signals with one another - in contrast to structurable logic arrangements which can be used universally.

The tasks which have to be performed by the respective sub-blocks, and the cross-connections between the individual sub-blocks make it possible for the relevant logic blocks to have similar performance to that of logic blocks constructed using structurable logic arrangements which can be used universally.

This holds true even when the logic blocks are divided into sub-blocks which are constructed differently and/or have to perform different tasks and/or have different cross-

connections and/or are provided in a larger or smaller number than the sub-blocks 5 to 8 which are used in the example considered in the present case and are described with reference to Fig. 4.

5

In the case of data group operations, it may be necessary to additionally integrate memory units into the SLE layer 12. In that case, results required for later calculations do not have to be swapped into storage devices provided outside the SLE layer 12, but rather can be immediately (buffer-) stored internally within the SLE layer 12.

In the example considered, the structurable logic arrangements used in the sub-blocks are DNF logic arrangements whose construction essentially corresponds to the construction of the DNF logic arrangements used in the clock generation unit 121; in this case, the input and output signals are not, of course, clock signals but rather data and/or control signals, and required clock signals (for example for clocking the flip-flops) are selected from the clock signals generated by the clock generation unit 121.

However, there is no restriction to the effect that the structurable logic arrangements of the logic block 122 are such or similar DNF logic arrangements; instead, it is also

possible to use NAND arrays, multiplexer-based variants or other structurable logic arrangements.

The configuration of the configurable elements of the SLE layer, i.e. the configuration of the multiplexers, the configurable connections within the structurable logic arrangements and the registers can essentially be effected like the configuration of the known field-programmable logic arrangements (PLAs, GALs, PLDs, FPGAs etc.).

A first possibility in this respect consists in the (irreversible) production or erasure of connections using so-called fuses or antifuses.

Another possibility consists in carrying out reversible configuration based on data representing the desired configuration, the data being stored in EPROMS, EEPROMS or the like provided inside or outside the program-controlled unit. As a result, the configuration of the program-controlled unit can be changed a limited number of times.

A further possibility consists in carrying out reversible configuration based on data representing the desired configuration, but where the data are stored in a RAM or the like. As a result, the configuration of the program-controlled

unit can be changed an unlimited number of times and very rapidly.

The EPROMs, EEPROMs, RAMs or other memories in which the data
5 representing the configuration of the program-controlled unit
are stored are preferably designed in such a way that they can
be inserted into the memory or I/O area which can be accessed
by the μ P core 11. The configuration of the SLE layer 12 can
then be set as desired by the μ P core 11 itself.

0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207

If the configuration or a reconfiguration of the SLE layer 12 is permitted at any time, then these tasks can even be swapped dynamically into the SLE layer.

5

Irrespective of this, that is say even when dynamic configuration of the SLE layer 12 is not possible, the loading on the μ P core 11 can be considerably relieved by the SLE layer 12. This is explained below using a practical example.

The example relates to analog/digital conversion of data. More precisely, suppose that

- an A/D converter (for example the A/D converter 15) having a conversion width of 8 bits is started by a timer (for example by the timer 17),
- the result of the A/D conversion is stored together with a 12-bit counting marker,
- the result of the A/D conversion is monitored in respect of specific limit values being exceeded and undershot, in which case branching to a specific routine should be effected when the limit values are exceeded or undershot,

20

25

- and the operation is terminated after 2048 measurements.

An application of this type requires a relatively high complexity in the case of a purely software-based solution.

5 Since a typical A/D converter integrated in a microcontroller does not yield the result spontaneously, that is to say works as a so-called flash converter and has a conversion time in the region of a few microseconds, one of the following paths must be taken in connection with the exact execution of the application specification:

1) The timer triggers an interrupt. The interrupt service routine starts the A/D conversion and is then ended. The A/D converter likewise triggers an interrupt when the conversion is ended. The interrupt service routine that is
5 thereupon executed causes the A/D conversion result to be read out and processed.

2) The timer triggers an interrupt. In the interrupt service
20 routine that is thereupon executed, the A/D conversion is started, there is a wait for the end of the conversion, and finally (after the end of the conversion) the A/D conversion result is read out and processed.

25 If the conversion times are shorter than the interrupt latency times, the second variant should be preferred. Otherwise, the

first variant would be preferable. In general, however, the third variant below is the most favorable:

3) The timer triggers an interrupt. In the interrupt service routine that is thereupon executed, the last A/D conversion result is read out, the next A/D conversion is started, and the A/D conversion result that has been read out is evaluated.

In this third variant, although the A/D conversion result is generally read and evaluated later than would actually be possible, this delay is generally tolerable.

The complexity that has to be implemented for practical realization of the third variant is undoubtedly the least. Nevertheless the processing of the interrupt service routine whose execution is initiated by the timer interrupt lasts at least (i.e. if one instruction is executed per clock cycle and if there are no instances of the limit values being exceeded or undershot by the A/D conversion results, and if there were not interrupt latency times) 30 clock cycles.

If the execution of this application is shifted as far as possible into the SLE layer 12, then it can be carried out considerably faster.

The logic block which is shown in Fig. 4 and described with reference thereto should then be configured, for example, in such a way that the structure shown in Fig. 5 is produced.

5 The input signals of the arrangement shown in Fig. 5 are the A/D conversion results of the A/D converter 15, which are designated by `ad[0..7]`, and also the pulses of the timer 17, which are designated by `Start_T`.

10 The structurable elements of the logic block in accordance with Fig. 5 are configured in such a way that said logic block, in the event of the limit values being exceeded or undershot by the A/D conversion results and when the end of the measurement series is reached, sends interrupt requests designated by `Irq_Con` and, for the purpose of storing the A/D conversion results together with the counting marker, injects corresponding instructions into the instruction pipeline of the μ P core. In this case, it is incumbent

- 20 ▪ upon the second sub-block 6 to effect central sequence control within the relevant logic block,
- upon the first sub-block 5 to effect comparison of the A/D conversion data `ad[0..7]` received from the A/D converter 15
- 25 with values `u[0..7]` (upper limit value) and `l[0..7]` (lower

limit value) stored in the constant register 53, and to effect generation of the interrupt requests Irq_Con,

- upon the third sub-block 7 to effect comparison of addresses
5 (for the purpose of completing the instructions to be injected and for the purpose of determining the end of the measurement series), and
- upon the fourth sub-block 8 to effect generation and/or
10 compilation and injection of the instructions to be injected.

In this case, the structurable logic arrangements 61 and 62 are configured in such a way that they generate output signals Irq_Gen, Inject_1, Inject_2, New_Adr and Read on the basis of
15 input signals MClk, Start_T, End_Of_Adr and Reset using internal signals S0, S1, S2 and End_Reached, as defined by the Boolean equations below:

20 In the structurable logic arrangement 61:

$$\begin{aligned}
 S0 &= /S2 * /S0 * Start_T * /End_Reached + \\
 &\quad /S2 * S1 * /S0 * /End_Reached; \\
 S1 &= /S2 * /S1 * S0 * /End_Reached + \\
 25 \quad &\quad /S2 * S1 * /S0 * /End_Reached; \\
 S2 &= /S2 * S1 * S0;
 \end{aligned}$$

```

S0.CLK = MClk;          /* Clocking with MClk */
S1.CLK = MClk;          /* Clocking with MClk */
S2.CLK = MClk;          /* Clocking with MClk */

/* This describes the pass through 5 states, triggered by
5 Start_T */

End_Reached = End_Of_Adr + End_Reached * /Reset;
End_Reached.CLK = MClk;    /* Clocking with MClk */

/* Stored End-Flag */

```

In the structurable logic arrangement 62:

```

New_Adr = /S2 * S1 * /S0 + /S2 * S1 * S0;

/* Signal for state 2 and 3 */

Inject_1 = /S2 * S1 * /S0;

/* Injection 1. Move instruction */

Inject_2 = /S2 * S1 * S0;

20 /* Injection 2. Move instruction */

Read = /S2 * /S1 * S0;

/* Triggering of the read operation */

25 Irq_Gen = End_Reached;

/* Triggering interrupt request at end */

```


The structurable logic arrangements 54 and 55 are configured in such a way that they generate output signals Irq_Con1, Irq_Con2, b_ad[0..7] on the basis of input signals ad[0..7], u[0..7], l[0..7], Read, Irq_Gen using internal signals tmp_upper [0..3], tmp_lower[0..3], tmp_eq[0..3] (for intermediate results), tmp_ad[0..7] (for buffer-storage after interrogation) and Irq_Tmp1 (for IRQ generation and resetting) as defined by the Boolean equations below:

In the structurable logic arrangement 54:

$$\begin{aligned} \text{tmp_upper}[z] &= /u[2z+1] * \text{ad}[2z+1] + \\ &\quad u[2z+1] * \text{ad}[2z+1] * /u[2z] * \text{ad}[2z] + \\ &\quad u[2z+1] * \text{ad}[2z+1] * /u[2z] * \text{ad}[2z]; \\ \text{tmp_lower}[z] &= l[2z+1] * /\text{ad}[2z+1] + \\ &\quad l[2z+1] * /\text{ad}[2z+1] l[2z] * /\text{ad}[2z] \\ &\quad /l[2z+1] * /\text{ad}[2z+1] l[2z] * /\text{ad}[2z]; \\ \text{tmp_eq}[z] &= u[2z+1] * \text{ad}[2z+1] * u[2z] * \text{ad}[2z] + \\ &\quad u[2z+1] * \text{ad}[2z+1] * /u[2z] * \text{ad}[2z] + \\ &\quad /u[2z+1] * \text{ad}[2z+1] * u[2z] * \text{ad}[2z] + \\ &\quad /u[2z+1] * /\text{ad}[2z+1] * /u[2z] * /\text{ad}[2z]; \end{aligned}$$

/* z runs from 0 to 3, comparison for larger/smaller and equality */

```

tmp_upper[z].CLK = /Read; /* clocking at read-out end */
tmp_lower[z].CLK = /Read; /* clocking at read-out end */
tmp_eq[z].CLK = /Read; /* clocking at read-out end */

```

```

5      tmp_ad[x] = ad[x]          /* x from 0 .. 7 */
      tmp_ad[x].CLK = /Read;      /* clocking at read-out end */

      Irq_Tmpl = 1;
      Irq_Tmpl.CLK = Irq_Gen; /* clocking with Irq_Gen */
10     Irq_Tmpl.AR = IrqlReset; /* reset with IrqlReset */
      /* Generating IRQ for take-up end */

```

In the structurable logic arrangement 55:

```

15     Irq_Con1 = Irq_Tmpl;
      /* IRQ is generated for take-up end */

      Irq_Con2 =      tmp_upper[3] + tmp_tmp[3] * tmp_upper[2] +
                      tmp_eq[3] * tmp_eq[2] * tmp_upper[1] +
20     tmp_eq[3] + tmp_eq[2] * tmp_eq[1] *
                      tmp_upper[0] +

                      /* Overflow */
                      tmp_lower[3] + tmp_eq[3] * tmp_lower[2] +
                      tmp_eq[3] * tmp_eq[2] * tmp_lower[1] +
25     tmp_eq[3] * tmp_eq[2] * tmp_eq[1] *
                      tmp_lower[0];

```

```
/* Underflow */
```

```
b_ad[y] tmp_ad[y]          /* y runs from 0 .. 7 */
```

The structurable logic arrangements 74 and 75 are configured in such a way that they generate output signals End_Adr and BA[0..15] on the basis of input signals A[0..15], EA[0..15] using internal signals temp[0..7] as defined by the Boolean equations below:

In the structural logic arrangement 74:

```
temp[x] = /A[2x] * /EA[2x] * /A[2x+1] * /EA[2x+1] +
          A[2x] * /EA[2x] * /A[2x+1] * /EA[2x+1] +
          A[2x] * /EA[2x] * A[2x+1] * EA[2x+1] +
          A[2x] * EA[2x] * A[2x+1] * EA[2x+1];
/* x runs from 0 to 7, comparison for equality */
```

In the structurable logic arrangement 75:

```
End_Adr = temp[0] * temp[1] * temp[2] * temp[3] *
          temp[4] * temp[5] * temp[6] * temp[7];
/* produces 1 if addresses and comparison address
   identical */
```

```
BA[y] = A[y];          /* y runs from 0 .. 15 */
```

The structurable logic arrangement 82 is configured in such a way that it generates output signals Inject and MP[0..31] (assuming 32-bit interface to the μ P core) on the basis of input signals Inject_1, Inject_2, BA[0..15], b_ad[0..7], and CR4[0..15] and defined by the Boolean equations below:

```

Inject = Inject_1 + Inject_2;
MP[0..15] = CR4[0..15] * Inject_1 +
            CR4[0..15] * Inject_2;
/* the 16-bit instruction code stored */
/* in CR4[0..15] is communicated */
MP[16..23] = BA[0..7] * Inject_1 +
            b_ad[0..7] * Inject_2;
MP[24..31] = BA[8..15] * Inject_1;
/* the operand or operands */
/* (16-bit address or 8-bit AD value) */
/* are communicated */

```

As is apparent from the explanations above, each time A/D conversion data from the A/D converter are read out and processed, the arrangement in accordance with Fig. 5 passes through four states, namely:

- 1) Reading of the A/D conversion result ad[0..7] from the A/D converter with automatic restart of the A/D converter.

2) The A/D conversion result is processed, i.e. the A/D conversion result is compared with an upper limit value $u[0..7]$ stored in the constant memory 53 and a lower limit value $l[0..7]$ likewise stored in the constant memory. At the same time, a first move instruction is injected into the pipeline of the μP core.

3) A second move instruction is injected. This instruction comprises a transfer of the A/D conversion result to the address that has been incremented in the meantime. If the A/D conversion result lies outside the range defined by the limit values, an interrupt request is simultaneously triggered.

4) The end of the 2048 conversions is possibly reached. In this case, a flag is set which prevents further reading-out and processing of A/D conversion results. Furthermore, a further interrupt request is triggered for the purpose of signaling the end of the routine.

If it is assumed that each state requires precisely one sequence clock signal, then 4 clock signals are required for each instance of reading out and processing A/D conversion data from the A/D converter; the μP core is temporarily burdened by

two or three CPU clock cycles through the injected instructions.

With incorporation of the SLE layer 12, the reading and
5 evaluation of A/D conversion results considered in the present
case can be carried out in a small fraction of the time which
would be necessary if the procedure were as it has been
heretofore, that is to say if the complete sequence control
and data processing were essentially left exclusively to the
10 μ P core 11.

It goes without saying that similar advantages can also be
obtained in conjunction with completely different applications
from the example described above.

15 As a result, the program-controlled unit described can be used
optimally in each case for a large number of highly varied
applications.